Computers and Mathematics with Applications 63 (2012) 588–600

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

# Towards efficient and practical network coding in delay tolerant networks☆

Baokang Zhao [a], Wei Peng [a], Ziming Song [b], Jinshu Su [a,*], Chunqing Wu [a], Wanrong Yu [a], Qiaolin Hu [c]

[a] *School of Computing Science, National University of Defence Technology, Changsha, Hunan, 410073, China*

[b] *Department of Electronics, Shenyang Institute of Artillery, Shenyang, Niaoning, 110016, China*

[c] *Department of Electronic Countermeasure, Air Force Radar Academy, Wuhan, Hubei, 430000, China*

ARTICLE INFO

*Article history:*
Received 30 April 2011
Received in revised form 2 October 2011
Accepted 3 October 2011

*Keywords:*
Network coding
Computer networks
Pervasive computing

ABSTRACT

Network coding techniques offer an emerging solution to efficient data transmission in Delay Tolerant Networks (DTN). To date, abundant techniques have been developed on exploiting network coding in DTN, however, most of them bring additional overhead due to the extra coded message redundancy. In this paper, we analyze the coded message redundancy issue, and then propose NTC, an efficient network coding scheme for DTN. In NTC, a novel metric named "redundancy ratio" is introduced within the anti-entropy message exchange process. We also discuss the design and implementation of practical NTC in detail. To evaluate the performance of our proposed NTC scheme, we implement NTC in ONE, the current state-of-the-art simulator for DTN. Simulation results show that, comparing with existing schemes, our proposed NTC scheme has significant advantages in enhancing the message delivery ratio and reducing the transmission overhead.



## 1. Introduction

Over the past few years, Delay Tolerant Networking (DTN) [1,2] has attracted a lot of research effort since it provides a feasible solution to efficient communication in delay/disruption environments. In DTN, the efficiency of data transmission is one of the most critical issues. Towards this issue, several schemes were proposed. In general, these schemes can be classified into two categories, i.e., the schemes with network coding and without network coding. Meanwhile, it is worth noting that network coding based approaches were more efficient and thus attracted more attention.

To enable practical network coding in delay tolerant networks, random linear network coding plays an important role due to its simplicity and efficiency. For the sake of convenience, in this paper, we use the term "network coding" to indicate the term "random linear network coding".

However, although most current network coding schemes are very helpful in data transmission, they also bring high transmission overheads due to redundant coded messages. In this paper, we analyze the problem of redundant coded messages, and propose a novel network coding-based data Transmission Control scheme (NTC) to reduce the overhead. In general, the contribution of this paper can be summarized as follows,

☆ Part of this work has been appeared in the conference proceedings of IMIS-2011.

* Corresponding author.
*E-mail addresses:* bkzhao@nudt.edu.cn (B. Zhao), wpeng@nudt.edu.cn (W. Peng), songzim@gmail.com (Z. Song), sjs@nudt.edu.cn (J. Su), xixiwu2001@yahoo.com.cn (C. Wu), ywr@nudt.edu.cn (W. Yu), huqiaolin@gmail.com (Q. Hu).

0898-1221/$ – see front matter © 2011 Published by Elsevier Ltd
doi:10.1016/j.camwa.2011.10.001

- We identify the packet redundancy problem for practical network coding in delay tolerant networks, and propose a novel metric of "code generation redundancy ratio" to indicate coded message redundancy. To the best of our knowledge, we are the first to study such a problem.
- We propose NTC, a novel Network coding-based data Transmission Control scheme to exploit the metric of code generation redundancy ratio.
- We also implement the NTC scheme within the ONE simulator, and evaluate its performance. Simulation results show the efficiency of our proposed NTC scheme.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of related work. In Section 3, we analyze the problem of coded message redundancy for random linear network coding in DTN. Thereafter, we propose the NTC scheme along with its design details in Section 4. The performance of the NTC scheme is evaluated in Section 5. Finally, we conclude the paper in Section 6.

## 2. Related work

Early work on data transmission in DTNs does not use network coding methods. Well-known algorithms include Epidemic Routing (ER) [3], Spary and Wait [4], Spray and Focus [5], and so forth. The Epidemic Routing (ER) scheme is a flooding-based algorithm in which a network node will exchange every message in its buffer with the node it meets until they have the same messages in their buffers. ER can achieve a high message delivery ratio but is very resource-consuming, which is not practical in real applications. Spray and Wait, Spray and Focus use limited message copies to eliminate the high message overhead in ER. In the two algorithms, a limited number of message copies are injected into the network and delivered to other nodes until only one message is left. The left message will be transferred to the destination directly. A similar algorithm named SMART [6] only sprays messages to the so-called companions of the destination node so that the message delivery ratio can be improved. PROPHET (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [7], and Advanced PROPHET [8] make their routing decisions according to the characteristics of recursiveness and locality in node movements.

Recently, stimulated by the efficiency and promising benefits of network coding, people has devised some schemes with network coding for data transmission in DTNs. It is believed that the network coding based algorithms will be more efficient that those without network coding in DTNs because such schemes are more resilient to the challenging conditions in DTNs, like intermittent connections, long transfer delays, etc. The erasure codes [9,10] and rateless codes [11] are evaluated using traditional end-to-end encoding techniques in communication theory. At present, data transmissions in DTNs using network coding have got exciting results [12,5,13,14].

Although many schemes with network coding have been proposed to improve the efficiency of data transmission, most of them focus on encoding and decoding techniques. An important problem has not been thoroughly studied. That is, the number of messages using network coding will increase quickly with the spreading of messages in DTNs, which always causes serious redundant coded messages. Therefore, it is very crucial to handle the redundant coded message problem. In the remainder part of this paper, we will analyze this problem in the context of random linear coding, and propose an efficient and practical scheme named NTC to tackle this problem.

## 3. The coded message redundancy problem

### 3.1. Preliminaries of network coding

Network Coding was firstly proposed in 1999 [15]. Since then, it has been widely investigated in numerous applications. We would like to review some mathematical fundamentals as follows.

Assume $G = (V, E)$ represents a communication network, where $V$ and $E$ denote the node set and edge (or called channel) set respectively. For each node $T \in V$, let $\mathrm{In}(T)$ and $\mathrm{Out}(T)$ denote the incoming channel set and outgoing channel set respectively. $|\mathrm{In}(T)|$ and $|\mathrm{Out}(T)|$ represent the size of set $\mathrm{In}(T)$ and $\mathrm{Out}(T)$. In network coding, a data unit is always represented by an element of a base field $F$, consequently, a message consisting of $\omega$ data units can be represented by an $\omega$-dimensional row vector $x \in F^{\omega}$.

Network coding is always specified as encoding mapping, where the mapping function usually maps the symbols received from all the incoming channels to a symbol of each outing channel. These mappings include local encoding mapping, global encoding mapping, etc.

**Definition 1** (*Local Encoding Mapping*)**.** Let $F$ be an infinite field and $\omega$ be a positive number, for each channel $e \in \mathrm{Out}(T)$ of node $T$ in $G = (V, E)$, its local encoding mapping can be denoted as,

$$\tilde{k}_e : F^{|In(T)|} \to F. \tag{1}$$

The local encoding mapping represents an $\omega$-dimensional mapping function for node $T$, where $\omega = |\mathrm{In}(T)|$ incoming messages will be mapped to a new message on the channel $e$.

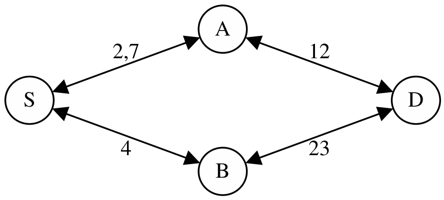


**Fig. 1.** An example of a space–time graph for DTN.

**Definition 2** (*Global Encoding Mapping*)**.** Let $F$ be a infinite field and $\omega$ be a positive number, for each channel $e \in \mathrm{Out}(T)$ of node $T$ in $G = (V, E)$, its global encoding mapping $\tilde{f}_e : F^\omega \to F$ is uniquely determined by $(\tilde{f}_d(x), d \in \mathrm{In}(T))$, and $\tilde{k}_e$ is the mapping via

$$(\tilde{f}_d(x), d \in \mathrm{In}(T)) \to \tilde{f}_e. \tag{2}$$

The global encoding mapping represents a global view of the mapping function for channel $e$, from a network level perspective.

Since our paper is focused on random linear network coding, we would like to introduce related concepts to random linear network coding.

**Definition 3** (*Linear Local Encoding Kernel*)**.** Let $F$ be an infinite field and $\omega$ be a positive number, regarding each node $T \in G = (V, E)$, let $(d, e)$ be a neighborhood edge of $T$ where $d \in \mathrm{In}(T)$ and $e \in \mathrm{Out}(T)$, $K_{d,e}$ is a linear local encoding mapping on the edge $(d, e)$, then, a linear local encoding kernel of node $T$ can be denoted as,

$$K_T = |\tilde{k}_{d,e}|_{d \in \mathrm{In}(T), e \in \mathrm{Out}(T)}. \tag{3}$$

The linear local encoding kernel represents a linear matrix to perform linear transform to incoming messages. From a specific node $T$, it will perform linear matrix transform on each incoming message with its linear local encoding kernel, and sends the results to corresponding output channel.

**Definition 4** (*Linear Global Encoding Kernel*)**.** Let $F$ be an infinite field and $\omega$ be a positive number. An $\omega$-dimensional $F$-valued linear network code consists of a scalar $\tilde{k}_{d,e}$ for each adjacent pair $(d, e)$, and the linear global encoding kernel of each channel $e$ can be denoted as,

$$f_e = \sum_{d \in \mathrm{In}(T)} \tilde{k}(d, e) f_d, \quad \text{where } e \in \mathrm{Out}(T). \tag{4}$$

It is a $\omega$-dimensional column vector, and represents a global view of the mapping function for channel $e$, from a network level perspective.

### *3.2. Network coding in DTN*

In DTN, since the node contact opportunities are probabilistic, the network topology changes frequently. Consequently, the space–time graph model [16] was adopted to describe the topology of DTN. In the space–time graph model, a vertex represents a common node in the network, while an edge denotes the time when a contact between its connection nodes may occur. For instance, in Fig. 1, since node S and node A have an opportunity to contact each other at time 2 and 7, a edge marked with "2, 7" is used to indicate the occupied time for the contact.

In network coding enabled DTN, a node usually encodes some messages inside its buffer (called "code generation") using coding parameters (called "coding coefficients"), and sends the coded messages to the next hop node during their connection. An example is shown in Fig. 2. Assume two messages $M_1, M_2$ should be sent from source node S to destination node D. At $t = 12$, as shown in subfigure (d), node A encodes the messages $C_1$ and $C_2$ into a new message $C_{1,3}$ and sends the coded message to node D. Thereafter, at $t = 23$, as shown in subfigure (e), node B sends $C_2$ to D. Finally, node D receives two coded messages $C_{1,3}$ and $C_2$, and D is able to decode original messages $M_1, M_2$. Therefore, by adopting network coding, a DTN network can achieve a higher message delivery ratio since it is able to transmit more coded messages during multiple time varying paths.

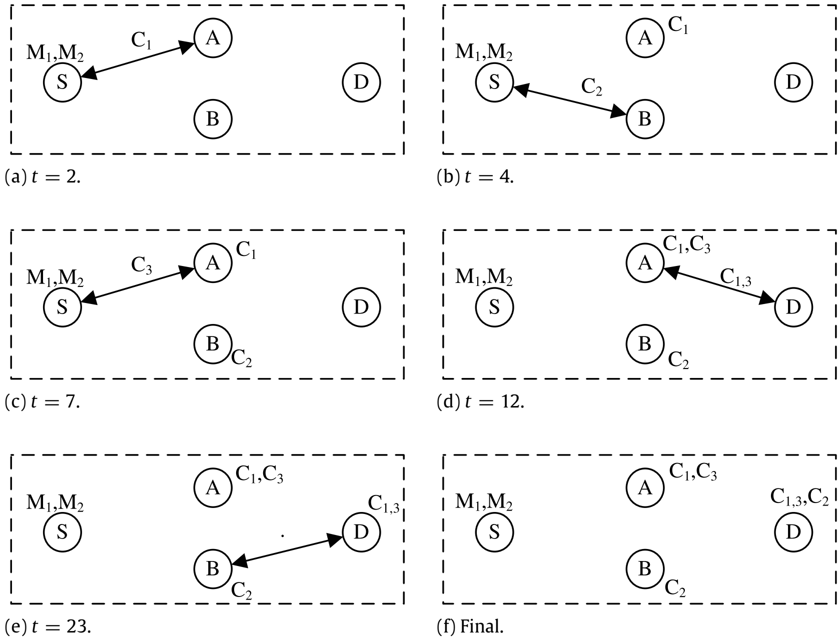


**Fig. 2.** An example of using network coding for data transmission in DTN.

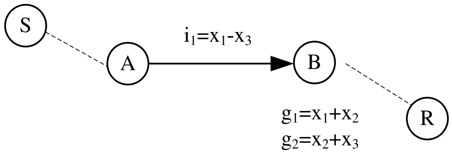


**Fig. 3.** A message redundancy example.

### *3.3. The coded message redundancy problem*

In practical network coding enabled DTN, after message encoding, a new id will be assigned to identify this new message. Therefore, together with the encoding process, the number of messages in the networks will grow rapidly. Accordingly, the overhead of message encoding, storage and communication will increase dramatically.

Let us look at an example. As shown in Fig. 3, let $S$ and $R$ denote the sender and receiver, $A$ and $B$ represent the intermediate nodes trying to exchange a coded message, respectively. Assume $B$ has a coded message $g_1$, $g_2$ represents the code generation of $R$ where $g_1 = x_1 + x_2$ and $g_2 = x_2 + x_3$, and $A$ is trying to send a coded message with $i_1 = x_1 - x_3$. Since $i_1$ is not linearly independent to $g_1$ and $g_2$, it is a redundant coded message for $B$, thus, the message processing overhead will be reduced if $A$ can avoid the redundant transmissions for message $i_1$.

This is a serious problem for practical network coding enabled transmission in DTN. In this paper, we propose NTC, a new scheme to provide a solution to the message redundancy problem.

## 4. The proposed NTC scheme

NTC is based on a novel metric named "redundancy ratio" in cooperation with the anti-entropy procedure.

### *4.1. The anti-entropy procedure*

In DTN, nodes usually exchange messages using Epidemic Routing [3], where a pair of nodes will execute an anti-entropy procedure [17] by exchanging a summary vector of the messages before real message transmission.

NTC also adopts the anti-entropy session together with network coding, as shown in Fig. 4. During this session, neighborhood nodes exchange their summary vectors, which usually present the meta data of their own messages, to determine messages required to be exchanged. One node (sender) initiates the session by sending its summary vector ($SVC_S$) to the other node (receiver). Then the receiver computes $SVC_S \wedge \neg SVC_R \wedge isUseful$, where $SVC_R$ denotes its summary vector, and *isUseful* is a Boolean vector to indicate whether messages in $SVC_S \wedge \neg SVC_R$ should be sent or not. Thereafter, the receiver

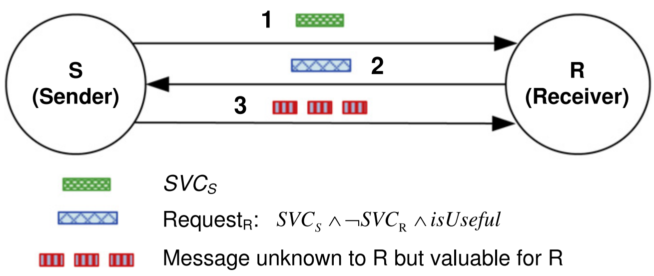


**Fig. 4.** A network coding-based anti-entropy session.

notifies the sender the result of $SVC_S \wedge \neg SVC_R \wedge isUseful$, and the sender transmits the related messages. On the reverse direction, the receiver also performs a similar process.

### *4.2. The metric of redundancy ratio*

To reduce the computational overhead, it is better if that, during coding, a node can evaluate the value of each message to be encoded. In practical network coding, messages that will be coded to the same message are always named as "*code generation*" [18], and the number of such messages is called "the rank of generation". The original messages represent the raw messages from source nodes within the network, and in practical linear network coding, these kind of messages are usually identified as noncoded messages and the identification only occupies one bit.

In NTC, a novel metric named "redundancy ratio" is proposed to measure the relationship between the number of original messages and the number of coded messages within a generation. Let #(originalMessages) and generationRank denote the number of original messages and coded messages for a generation, respectively, the redundancy ratio of this generation can be represented as,

$$\text{redundancyRatio} = \frac{\text{generationRank}}{\#(\text{originalMessages})}. \tag{5}$$

The "redundancy ratio" can be used to control the number of the coded messages under a specific redundancy ratio.

### *4.3. The NTC encoding procedure*

The basic idea of NTC is to stop encoding the new coming message if this new message is unable to increase the information of the new generation. Assume node $A$ is preparing to send messages to node $B$, and $CSV$ denotes the summary vector, the NTC encoding procedure can be summarized as shown in Algorithm 1.

**Algorithm 1: NTCEncoding**

**Input**: $CSV_B$: the summary vector of a generation received from sender $B$
**Output**: Whether the node needs to request a coded message
Receive a summary vector $CSV_B$ from sender $B$;
**for** *each summary vector* $sv_m$ *in* $CSV_B$ **do**
  **if** $(sv_m \in CSV_A Gen) == FALSE$ **then**
    return TRUE;
  **end**
  **else if** $(CSV_A.Gen.codedMsgs.size \geq (CSV_A.Gen.originalMsgs.size \times rRatio))$ **then**
    return FLASE;
  **end**
  return TRUE;
**end**

In this procedure, the redundancy ratio *rRatio* is used to control the number of redundant messages, and thus avoids unnecessary transmission.

**Theorem 1.** *Compared with Epidemic Routing, NTC is able to reduce message redundancy.*

**Proof.** In Epidemic Routing, a node needs to accept all the coded messages if it does not have the same generation. However, in NTC, a node firstly evaluates the redundancy of the message, and then stops encoding if the received message is unable to bring new messages into the current generation. Consequently, NTC should reduce message redundancy. □

Moreover, to evaluate the computational complexity, we have the following theorem.

| 8 bits | 8 bits | 8 bits | 8 bits |
|---|---|---|---|
| Version | Processing control flags(Variable Length) | | |
| Block size(Variable Length) | | | |
| Network coding head(Optional, Variable Length) | | | |
| Payload(Variable Length) | | | |

**Fig. 5.** The data message format of NTC.

| 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits |
|---|---|---|---|---|---|---|---|
| Gen.id | Dim | Msg.id | Msg.id | ⋯ | NC coef | NC coef | ⋯ |

**Fig. 6.** The network coding header for random linear coding.

**Theorem 2.** *Let $m$ and $n$ represent the number of generations in the buffer of the sender and receiver respectively, the computational overhead of NTC is smaller than $o(mn)$.*

**Proof.** The main computational overhead is the comparison of the summary vector of code generations during contact within two communication nodes. The complexity to determine whether each generation summary vector of the sender $s$ is included in the receiver's generation is $o(mn)$. Since in DTN, the number of messages is much larger than $m$ and $n$, therefore, the complexity of NTC is $o(mn)$. □

### *4.4. The design details of NTC*

The design of NTC consists of two parts, the control plane and data plane. The control plane defines the data packet format of coded messages, and the data plane includes the new message processing, encoding, and decoding procedures.

#### *4.4.1. The data packet format of coded messages*

In general, the messages in NTC can be classified as control messages and data messages. Since control messages in NTC are as normal as ordinary DTN control messages, we mainly focus on the format of data messages.

We define a general message format for NTC data message as shown in Fig. 5. The "version" field denotes the protocol version occupies one byte. The length of the "processing control flags" field is dynamic, and it contains many controlling flags, such as whether the custody transfer protocol is supported, whether acknowledgment is requested, etc. The field of "network coding header" is optional, and when the random network coding is enabled, it will be filled with the message id and related coefficients. The "payload" field is occupied by the content of the message, and thus, its length is flexible.

In Fig. 6, a coding header for random linear coding was shown as an example. In this header, Gen.id and Dim represents the id and dimension of generation, where Msg.id and NC.coef denotes the message id and coefficients, respectively.

#### *4.4.2. The procedure of new message processing*

When a source node creates a message, it first checks its buffer to see if the generation that the message belongs to exists. If it exists, then it encodes messages of the same generation and puts the coded message into the buffer. Otherwise, it creates a new generation and puts the new messages into it. The procedure is the same at the source node as with the relay node. The procedure is shown as Fig. 7.

#### *4.4.3. The procedure of message encoding*

The encoding procedure is shown in Fig. 8. The node first gets the messages in the generation. It then compares the number of the messages $m$ with the pre-defined constant $n$ that limits the maximum of messages to be encoded together. If $m \geq n$, we should randomly choose $n - 1$ messages from the generation and encode them with the new message in a finite field. Otherwise if $m < n$, we can directly combine the new message with the messages in the generation. The encoding coefficients are randomly chosen from a finite field. At last, put the coded message and other messages into the buffer of the node in the form of a generation.

#### *4.4.4. The procedure of message decoding*

The destination also organizes the coded messages it received into a generation. There is only one generation for a destination node when using the same destination encoding mechanism. The destination node first reduces the newly received coded message using the decoded original messages, then tries to perform a Gaussian Elimination procedure in order to convert the decoding matrix into a triangular one and eventually decodes all the original messages. The node puts

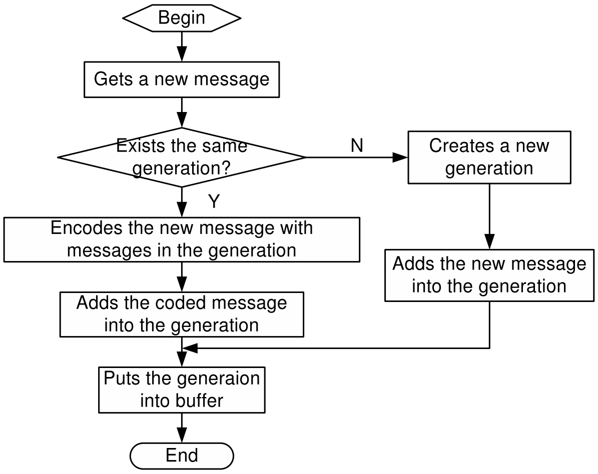


**Fig. 7.** The procedure of new message processing.

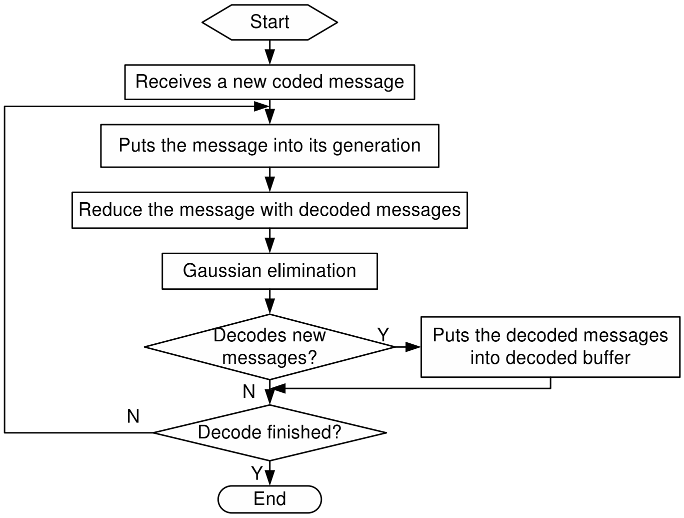


**Fig. 8.** The procedure of message encoding.

the decoded messages into the decoded buffer and acknowledges the successful reception of the original messages from the source node. It may take a little time until the destination can decode all the coded messages waiting to receive as many messages as needed. The entire procedure is shown in Fig. 9.

## 5. Performance evaluation

### 5.1. Simulation setup

To evaluate the performance of our proposed NTC schemes, we conducted extensive simulations on the ONE [19] simulator.

The main parameter settings are listed in Table 1. A few comments on the parameters are listed as follows.

(1) The transmission range of regular nodes is set to 100 m. It is a sparse network and the contact opportunities among nodes are low.

(2) As suggested in [14], some ferry nodes were added into the network to enhance the efficiency to all schemes, including epidemic routing, NTC, etc. the transmission range of ferries is the same as that of regular nodes.

(3) No special mobility model is imposed on ferries. Instead, we use the Random Walk mobility model built in ONE for ferries. It should be noted that the main advantage of ferries is their controllability on movement routes. In realistic deployments, the routes of ferries are often pre-computed in order to make full use of their advantages. Therefore, the simulation results using good-planned routes for ferries will exhibit high performance.

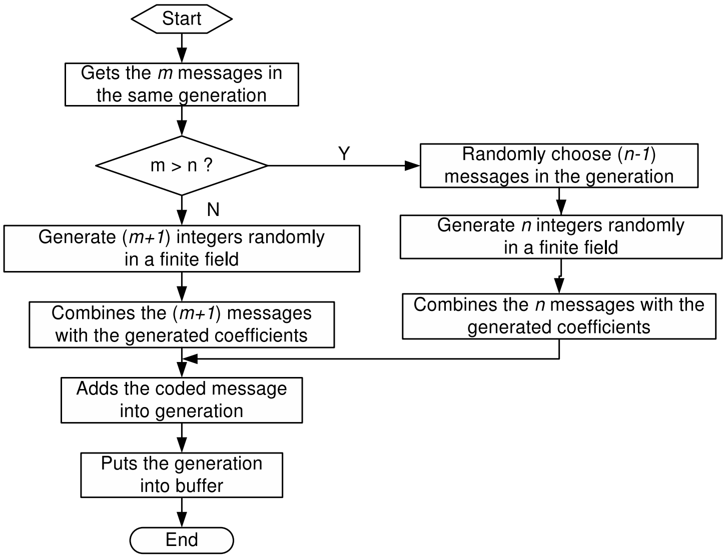


**Fig. 9.** The procedure of message decoding.

**Table 1**
List of simulation parameters.

| | | |
|---|---|---|
| Scenario | Scenario size | 5000 m * 5000 m |
| | Number of nodes | 20 |
| | Transmission model | Epidemic routing/NTC |
| | Movement model | Random way point |
| Node | Movement speed | Random from 5 m/s–10 m/s |
| | Transmit speed | 2 Mbps |
| | Buffer size | 5 MB |
| | Transmit range | 100 m |
| Ferry | Movement speed | 10 m/s |
| | Transmit speed | 1 Mbps |
| | Buffer size | 500 MB |
| | Transmit range | 500 m |
| | Movement model | Random walk |
| Message | Size | 500 kB–1 MB |
| | Number | 200 |
| | Creation event | Randomly created in 1000 s |
| | TTL | 30 min |

(4) For the convenience of comparison, the message events are configured statically. There are 200 unicast messages randomly generated in 1000 s, thus 100 messages for each node. The message size is distributed randomly from 500 kB to 1 MB. Because the buffer size is only 5 MB for each regular node, the network has a heavy payload.

### *5.2. Simulation results and performance analysis*

Based on the extended ONE simulator, we conducted extensive simulations. In the following results, all of the 200 messages are generated in the first 1000 s. Below, we will compare the simulation results and perform related analyses.

#### *5.2.1. The impact of redundancy ratio*

In the implementation, the number of coded messages in one network coding generation can be limited by the redundancy ratio. In general, a large redundancy ratio guarantees that there are enough coded messages in the network. It conduces decoding coded messages and recovering original messages in the destination nodes, thus improving the delivery ratio. On the other hand, when the buffer size of network nodes is limited, the buffer will be exhausted quickly due to the increase of coded messages caused by the high redundancy ratio. Some nodes have to delete some earlier messages in order to provide enough buffer space for new messages, which leads to reducing the number of messages for other generations, and brings adverse effects on delivery ratio.

Fig. 10 shows the variations of the delivery ratio and the transmission overhead with the redundancy ratio varying between 0.5 and 5 over a time scale of 2000 and 3000 s. As shown in the figure, as the redundancy ratio increases, the

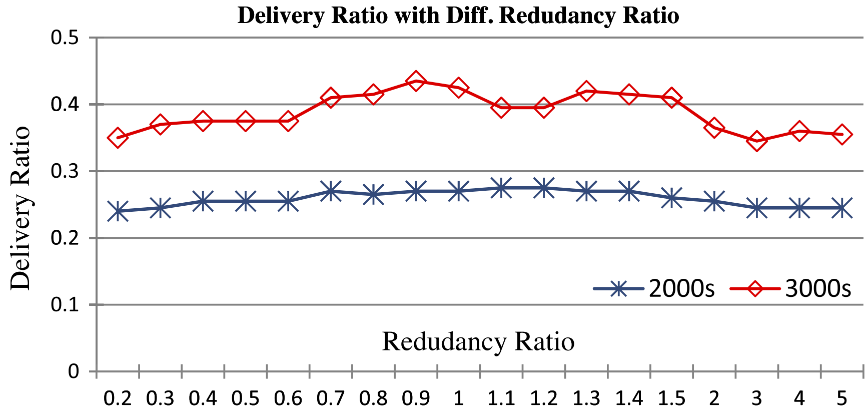


**Fig. 10.** Delivery ratio with diff. redundancy ratio.

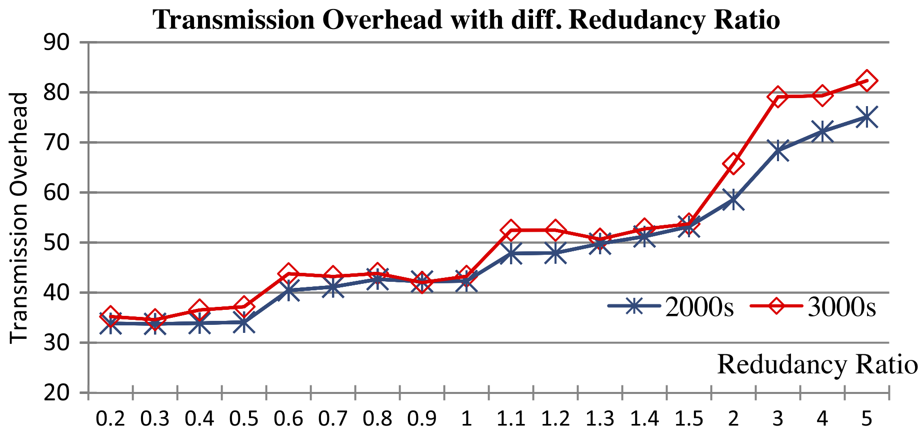


**Fig. 11.** Overhead with diff. redundancy ratio.

delivery ratio increases, whereas there is a downward trend in the delivery ratio when the redundancy ratio goes up to a certain extent because the buffer space of nodes is limited. As the redundancy ratio increases, the transmission overhead rises, because data transmission costs use more system resources when the redundancy ratio is larger.

Fig. 11 shows the variations of the delivery ratio and transmission overhead with the redundancy ratio over a longer simulation time. Under the current parameter settings, the impact of redundancy ratio to the delivery ratio is not significant. The delivery ratio is similar under different redundancy ratios. With regard to different simulation times, the impact of redundancy ratio is limited, too. The curves of transmission overhead show that bigger redundancy ratio will lead to greater transmission overhead over the network. We should take the transmission overhead into account when we want to gain a high delivery ratio through modifying the redundancy ratio. Figs. 12 and 13 show the delivery ratio and transmission overhead with simulation time, they also show the impact of redundancy ratio. Therefore, we need have a balance between delivery ratio and transmission overhead. Without a special statement in the follow-up simulations, the redundancy ratio is set to 1.0.

### *5.2.2. Performance comparison for network coding vs. epidemic routing*

Figs. 14 and 15 give the comparison of the delivery ratio and the transmission overhead between the network coding based scheme and epidemic routing. The redundancy ratio of network coding is set to 1.0 in this simulation.

We can see from Fig. 14 that the delivery ratio of the network increases when network coding is applied. At the beginning of the simulation, the delivery ratio of network coding is nearly the same as epidemic routing, even lower sometimes. The reason is that with network coding, it takes time to cumulate coded messages for decoding. The destination cannot decode messages when there are not enough messages.

With the simulation time going by, the delivery ratio of epidemic routing does not increase for two reasons. Firstly, some messages have reached their TTL values and been discarded in relaying nodes. Secondly, the buffer of the relaying node has been filled with message copies and cannot make room for new messages, thus leading to the result that the delivery ratio does not increase. On the contrary, when the network coding technique is used, although some messages may reach their

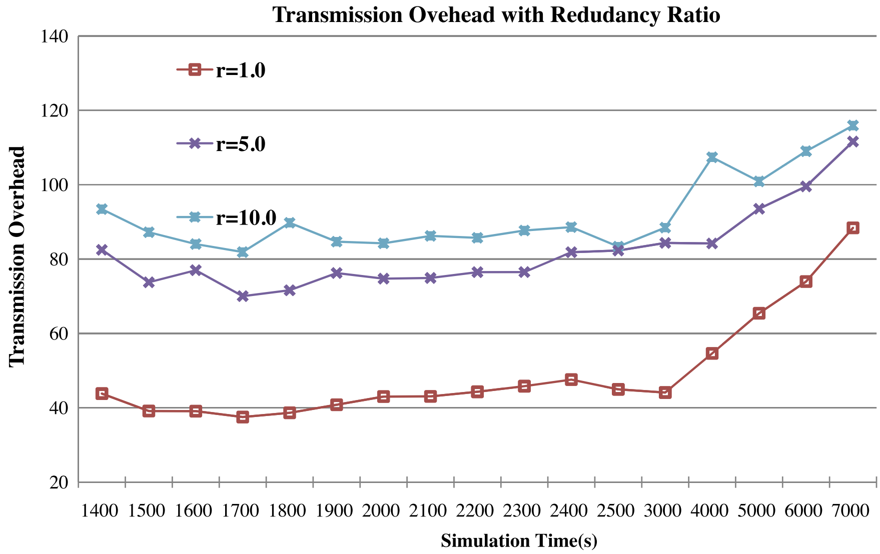


**Fig. 12.** Transmission overhead with diff. redundancy ratio.

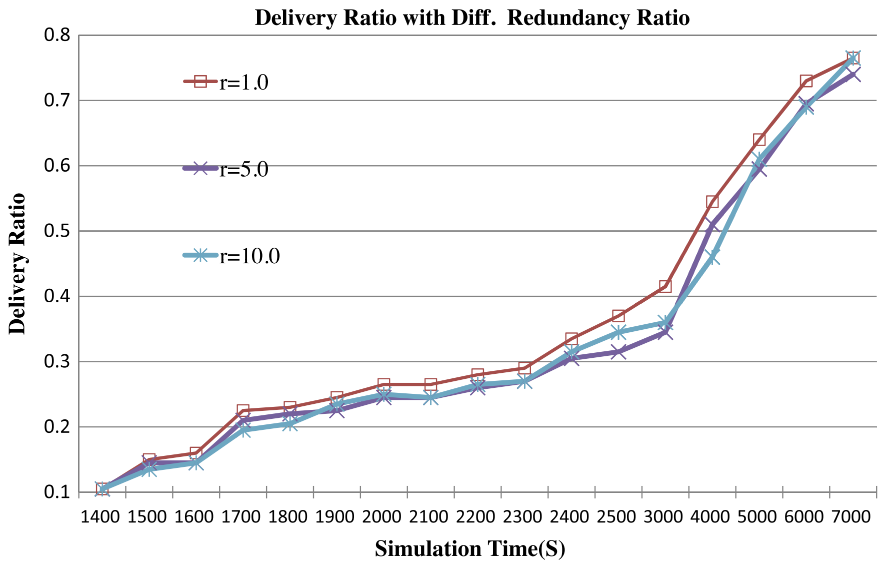


**Fig. 13.** Delivery ratio with diff. redundancy ratio.

TTL values and thus are deleted, other coded messages still contain the information of the original messages. The destination can recover the original messages using these coded messages. Thus the delivery ratio increases with the time going by.

The application of network coding increases the number of messages delivered in the network. Therefore, the transmission overhead ratio increases with increasing time. The results are shown in Fig. 15.

### 5.2.3. Performance analysis of NTC

We can see from the former theoretical analysis and experimental results that the transmission overhead increases quickly at the latter period of the simulation. We put forward the NTC algorithm to control the data transmission overhead. Fig. 16 gives the efficiency of NTC. The delivery ratio when using NTC approximates to the situation when using random network coding simply in DTN. The mechanisms in NTC will not reduce the delivery ratio for a large amount. Meanwhile, from Fig. 17, we can see that the transmission overhead is greatly reduced in NTC and is comparable to the epidemic routing. In summary, NTC can decrease the transmission overhead significantly without sacrificing the delivery ratio.

We also evaluate the performance of NTC with the number of ferries. Experimental results are listed in Figs. 18 and 19. As shown in Fig. 18, when there are more ferries, the delivery ratio is higher, since that ferries can have more opportunities to

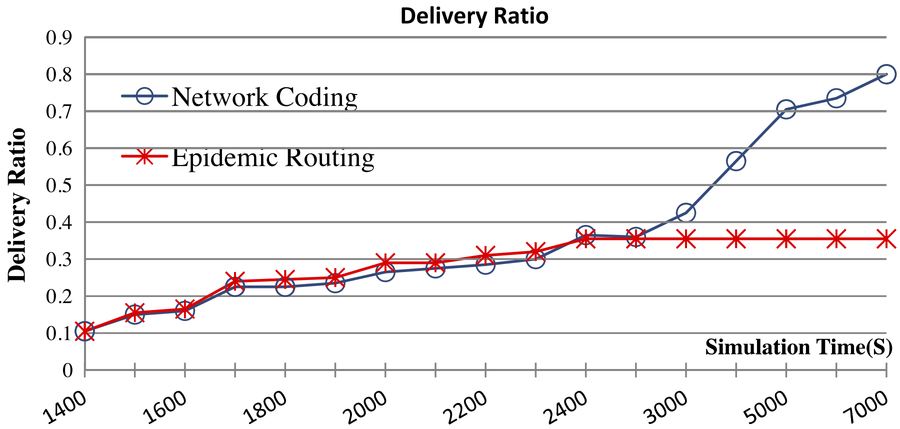


**Fig. 14.** The delivery ratio comparison.

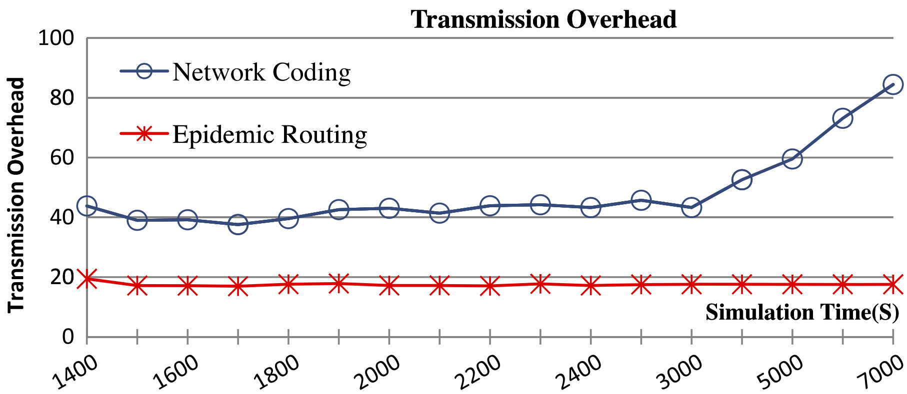


**Fig. 15.** The message overhead comparison.

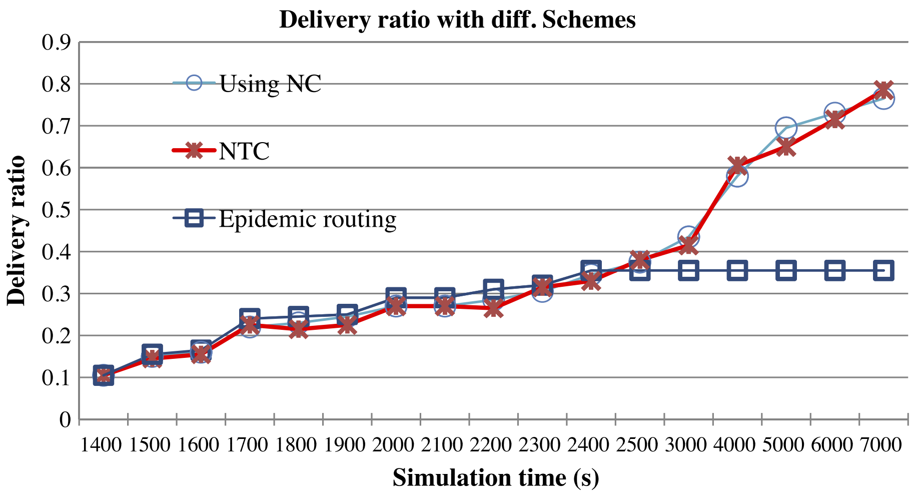


**Fig. 16.** Delivery ratio with diff. schemes.

communicate with other nodes. Moreover, as the time goes by, more ferries with the NTC scheme can significantly increase the delivery ratio. Also as shown in Fig. 19, more ferries means more transmission overhead, and NTC will bring more transmission overhead compared with non-coding schemes.

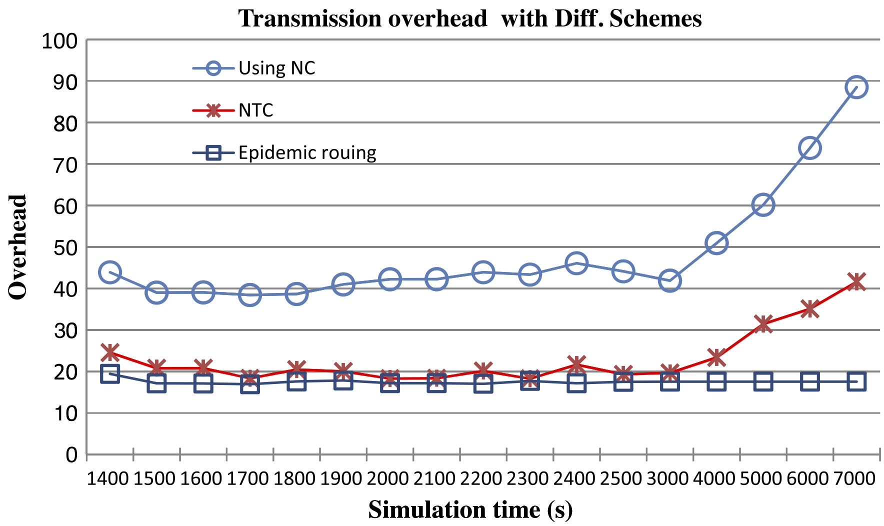


**Fig. 17.** Transmission overhead with diff. schemes.

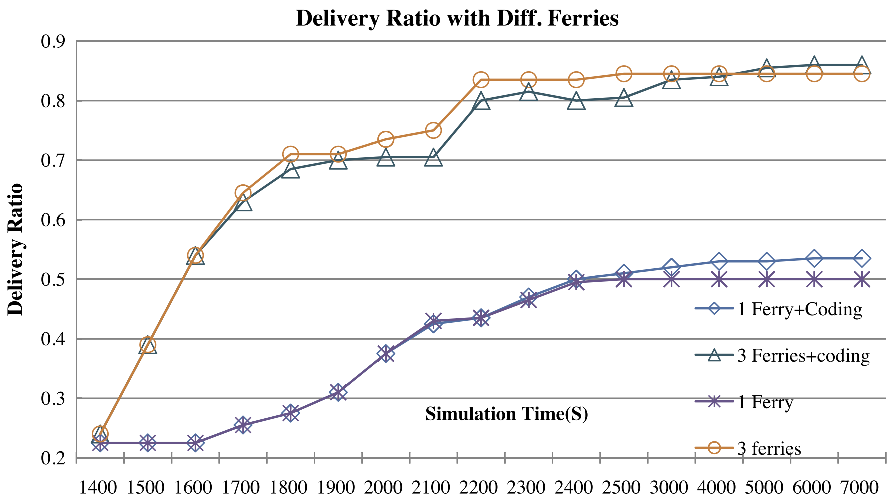


**Fig. 18.** Transmission overhead with different ferries.

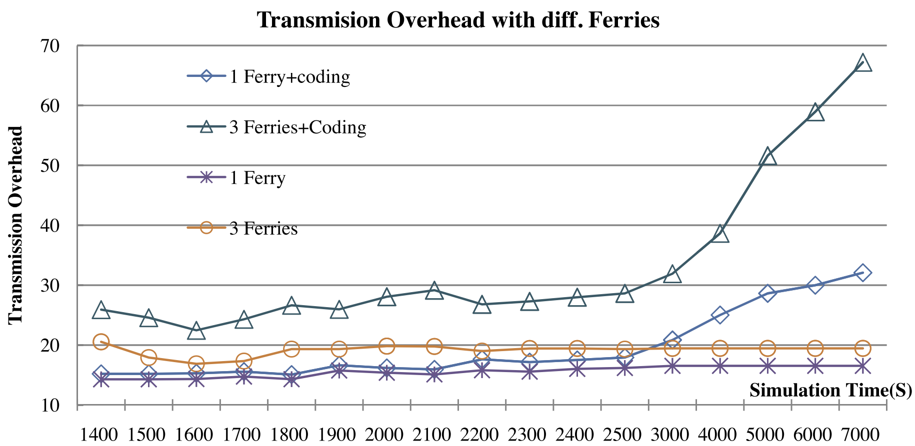


**Fig. 19.** Transmission overhead with different ferries.

## 6. Conclusion and future work

In this paper, we first addressed the coded message redundancy problem in network coding enabled DTN. Then, towards this end, we proposed NTC, an efficient network coding scheme for efficient message delivery in DTN. In NTC, to reduce the coded message redundancy, a novel metric named "redundancy ratio" was introduced within the anti-entropy message exchange process. We discussed the design and implementation of practical NTC in detail. Finally, we implemented NTC in the ONE simulator and conducted extensive simulations. Simulation results showed that, compared with existing schemes, our proposed NTC scheme has significant advantages in enhancing the message delivery ratio while reducing the transmission overhead.

As a first work for addressing the message redundancy problem with network coding in DTN, we provided a comprehensive study but focused more on the random linear network coding with unicast applications. We admit that there are many important problems yet to be studied. In the future, we will extend our research to multicast and anycast applications.

## Acknowledgments

The work described in this paper is partially supported by the grants of the National Basic Research Program of China (973 project) under Grant No. 2009CB320503, 2012CB315906; the National 863 Development Plan of China under Grant No. 2009AA01A334, 2009AA01A346, 2009AA01Z423; and the project of National Science Foundation of China under grant No. 61070199, 61003301, 60903223, 60903224, 61103194, 61103182; and supported by Program for Changjiang Scholars and Innovative Research Team in University of the Ministry of Education ("Network Technology", NUDT), the Innovative Research Team in University of Hunan Province ("Network Technology", NUDT), and the Innovative Research Team of Hunan Provincial natural science Foundation (11JJ7003).

## References

[1] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, SIGCOMM Comput. Commun. Rev. 34 (2004) 145–158.
[2] I.F. Akyildiz, Özgür B. Akan, C. Chen, J. Fang, W. Su, Interplanetary Internet: state-of-the-art and research challenges, Comput. Netw. 43 (2003) 75–112.
[3] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. Rep., Department of Computer Science, Duke University, 2000.
[4] J. Xue, X. Fan, Y. Cao, J. Fang, J. Li, Spray and wait routing based on average delivery probability in delay tolerant network, in: Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing—Volume 02, IEEE Computer Society, Washington, DC, USA, 2009, pp. 500–502.
[5] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOMW'07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 79–85.
[6] L. Tang, Q. Zheng, J. Liu, X. Hong, SMART: a selective controlled-flooding routing for delay tolerant networks, in: Proceedings of 4th International IEEE Conference on Broadband Communications, Networks, Systems, vol. 1, 2007, pp. 356–365.
[7] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, SIGMOBILE Mob. Comput. Commun. Rev. 7 (2003) 19–20.
[8] J. Xue, J. Li, Y. Cao, J. Fang, Advanced prophet routing in delay tolerant network, in: Proceedings of the 2009 International Conference on Communication Software and Networks, ICCSN'09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 411–413.
[9] S. Jain, M. Demmer, R. Patra, K. Fall, Using redundancy to cope with failures in a delay tolerant network, SIGCOMM Comput. Commun. Rev. 35 (2005) 109–120.
[10] J. Lacan, E. Lochin, Rethinking reliability for long-delay networks, in: Proceedings of 2008 International Workshop on Satellite and Space Communications, 2008, pp. 90–94.
[11] Y. Lin, B. Liang, B. Li, Performance modeling of network coding in epidemic routing, in: Proceedings of the 1st International MobiSys. Workshop on Mobile Opportunistic Networking, MobiOpp'07, ACM, New York, NY, USA, 2007, pp. 67–74.
[12] X. Zhang, G. Neglia, J. Kurose, D. Towsley, On the benefits of random linear coding for unicast applications in disruption tolerant networks, in: Proceedings of the 2006 IEEE Network Coding Workshop, 2006, pp. 1–7.
[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, Xors in the air: practical wireless network coding, SIGCOMM Comput. Commun. Rev. 36 (2006) 243–254.
[14] S. Ahmed, S.S. Kanhere, Hubcode: message forwarding using hub-based network coding in delay tolerant networks, in: Proceedings of the 12th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWiM'09, 2009, pp. 288–296.
[15] R. Ahlswede, N. Cai, S.Y.R. Li, R.W. Yeung, Network information flow, IEEE Trans. Inform. Theory 46 (4) (2000) 1204–1216.
[16] S. Merugu, M. Ammar, E. Zegura, Routing in space and time in networks with predictable mobility, Tech. Rep., Georgia Institute of Technology, 2004.
[17] D.B. Terry, M.M. Theimer, K. Petersen, A.J. Demers, M.J. Spreitzer, C.H. Hauser, Managing update conflicts in bayou, a weakly connected replicated storage system, SIGOPS Oper. Syst. Rev. 29 (1995) 172–182.
[18] P. Chou, Y. Wu, K. Jain, Practical network coding, in: Proceedings of the 2003 Conference on Communication, Control and Computing, 2003.
[19] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Simutools'09, ICST, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Brussels, Belgium, Belgium, 2009, pp. 1–10. ICST.